Journal of Computational and Applied Mathematics 290 (2015) 26–43

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Inexact Restoration method for nonlinear optimization without derivatives

Ma. Belén Arouxét [a], Nélida E. Echebest [a,*], Elvio A. Pilotta [b]

[a] Departamento de Matemática, FCE, Universidad Nacional de La Plata, CP 172, 1900 La Plata, Argentina

[b] Facultad de Matemática, Astronomía y Física, CIEM (CONICET), Universidad Nacional de Córdoba, Medina Allende s/n, Ciudad Universitaria (5000), Córdoba, Argentina

A R T I C L E   I N F O

Article history:
Received 29 April 2014
Received in revised form 23 March 2015

MSC:
65K05
90C30
90C56

Keywords:
Inexact Restoration
Derivative-free optimization
Trust-region methods
Polynomial interpolation

A B S T R A C T

A derivative-free optimization method is proposed for solving a general nonlinear programming problem. It is assumed that the derivatives of the objective function and the constraints are not available. The new method is based on the Inexact Restoration scheme, where each iteration is decomposed in two phases. In the first one, the violation of the feasibility is reduced. In the second one, the objective function is minimized onto a linearization of the nonlinear constraints. At both phases, polynomial interpolation models are used in order to approximate the objective function and the constraints. At the first phase a derivative-free solver for box constrained optimization can be used. For the second phase, we propose a new method ad-hoc based on trust-region strategy that uses the projection of the simplex gradient on the tangent space. Under suitable assumptions, the algorithm is well defined and convergence results are proved. A numerical implementation is described and numerical experiments are presented to validate the theoretical results.

Contents

* Corresponding author.
E-mail addresses: belen@mate.unlp.edu.ar (M.B. Arouxét), opti@mate.unlp.edu.ar (N.E. Echebest), pilotta@famaf.unc.edu.ar (E.A. Pilotta).

http://dx.doi.org/10.1016/j.cam.2015.04.047

1. Introduction

We present a new method for solving the general nonlinear programming problem

$$\min f(x) \text{ subject to } x \in \Omega, \quad C(x) = 0, \tag{1}$$

where $\Omega = \{x \in \mathbb{R}^n | L \leq x \leq U,\ L < U\}, f : \mathbb{R}^n \to \mathbb{R},\ C : \mathbb{R}^n \to \mathbb{R}^m$, where the derivatives of the objective function and the constraints are not available, although we assume that all the functions are continuously differentiable.

This kind of problems appears in many real world situations. For instance, when the functional values are the results of physical measurements or when the calculation of analytical derivatives is impractical [1–3].

Several methods and algorithms were developed for the unconstrained and box-constrained cases [4–9]. Later, in the last decade, some methods for the linearly constrained optimization problems without derivatives were proposed [10–15]. Derivative-free methods for more general constraints were addressed by means of Augmented Lagrangian approaches in [16–18].

Following the ideas of Powell's methods (BOBYQA) [9], where polynomial interpolation and trust-region strategy were used for box-constrained derivative-free optimization, we propose a method for the general optimization problem. Our method is based on the Inexact Restoration (IR) approach introduced in [19] and revised in [20,21]. A survey on this subject can be found in [22]. Each iteration includes two different phases: restoration and optimization. In the Restoration phase, which is executed once per iteration, an intermediate point (restored point) is found such that its infeasibility is a fraction of the infeasibility of the current point.

At the Optimization phase, a trial point belonging to π_k, a linearization of the feasible region around the restored point, is computed such that the objective function value is lower than in the restored point. A Lagrangian function can be also used at the Optimization phase as it is proposed in [23,21]. By means of a merit function, the new iterate is accepted or rejected. In case of rejection, the trust-region radius is reduced and the Optimization phase is repeated around the same restored point. This method improves almost separately the infeasibility and optimality. Filter criterion could be used instead of using a merit function [24–27]. One of the more attractive features of the IR method is that its theory allows us to use any efficient algorithm to perform each phase.

Recently, Bueno–Friedlander–Martínez–Sobral [28] also proposed a method based on IR for solving a nonlinear derivative-free optimization problem in which the derivatives of the constraints are available. In our work, the derivatives of the objective function and the constraints are not available and we approximate them by polynomial models, which is one of the main differences with the previous cited work.

We have taken into account the flexibility that IR method provides for choosing different subalgorithms in each phase, and therefore we performed two implementations of our method using two different solvers for Restoration phase: BOBYQA [9] and TRB-Powell [4].

On the other hand, for the Optimization phase, a derivative-free optimization problem with linear constraints is formulated. This problem could be solved by any efficient solver for linearly constrained derivative-free optimization, as the method introduced by Kolda, Lewis and Torczon [13], however we formulated an algorithm ad-hoc.

The paper is organized as follows. In Section 2 we briefly describe the IR method [19]. In Section 3 we introduce our derivative-free algorithm (IR-DFO) and some preliminary theoretical results. Also, we prove that the new algorithm is well defined. Assuming suitable hypotheses we analyze some global convergence results in Section 4. Implementation details and numerical experiments are shown in Section 5. Finally, some conclusions are made in Section 6.

Notation. Unless otherwise specified, our norm $\|\cdot\|$ is the standard Euclidean norm.

We let B denote a closed ball in $\mathbb{R}^n$ and $B(z; \Delta)$ denote the closed ball centered at z, with radius $\Delta > 0$.

e_i denotes the ith coordinate vector of $\mathbb{R}^n$.

We denote $C'(x) \in \mathbb{R}^{m\times n}$, the Jacobian matrix of $C(x)$ and $C_j'(x) = \nabla C_j(x)^T$ for $j = 1, \ldots, m$.

2. Inexact Restoration methods

In this section we give a description of the IR method [19] along with some preliminary definitions.

First of all, we define a measure of infeasibility given by: $h(x) = \|C(x)\|$. We used a penalty-like nonsmooth merit function, which combines feasibility and optimality, to measure the progress to the solution. This function is given by

$$\psi(x, \theta) = \theta f(x) + (1 - \theta)h(x), \tag{2}$$

where $\theta \in [0, 1]$ is a penalty parameter used to give different weights to the objective function and the measure of infeasibility. The choice of the parameter θ at each iteration depends on practical and theoretical considerations. See [19].

Given $y^k \in \mathbb{R}^n$ we define a linear approximation of the feasible region of (1) as

$$T(y^k) = \{x \in \Omega | C'(y^k)(x - y^k) = 0\}. \tag{3}$$

Moreover, given $z \in \mathbb{R}^n$ we also define $d_c(z)$ the projected direction of $-\nabla f(z)$ onto $T(z)$ as

$$d_c(z) = P_{T(z)}(z - \nabla f(z)) - z, \tag{4}$$

where $P_{T(z)}(w)$ denotes the orthogonal projection of w onto $T(z)$. A feasible point z such that $d_c(z) = 0$ is considered as a stationary point of (1) [29].

The IR model algorithm for solving (1) has the following form:

Inexact-Restoration model algorithm

Assume that $\alpha \in [0, 1)$, $\beta > 0$, $\delta_{\min} > 0$, $\theta_{-1} \in (0, 1)$ are algorithm parameters independent of k. Let $\{\omega_k\}$ be a summable sequence of nonnegative terms and $x^0 \in \Omega$. Set $k \leftarrow 0$.

Step 0: *Penalty parameter*

Define $\bar{\theta}_k = \min\{1, \min\{\theta_{k-1}, \ldots, \theta_{-1}\} + \omega_k\}$.

Step 1: *Restoration phase*

If $h(x^k) = 0$ we set $y^k = x^k$, otherwise compute $y^k \in \Omega$ such that

$$\|y^k - x^k\| \le \beta h(x^k) \tag{5}$$

and

$$h(y^k) \le \alpha h(x^k). \tag{6}$$

If this is not possible, stop the execution of the algorithm declaring *failure in improving feasibility*.

Step 2: *Optimization phase*

If $y^k = x^k$ and $d_c(x^k) = 0$ terminate the execution of the algorithm declaring *finite convergence*. Otherwise, choose $\delta \ge \delta_{\min}$.

Step 2.1: Compute $z^k \in T(y^k)$ as the solution of

$$\begin{array}{ll} \min & f(x) \\ \text{s. t.} & x \in T(y^k) \\ & \|x - y^k\| \le \delta. \end{array} \tag{7}$$

Step 2.2: Choice of the penalty parameter

Define, for all $\theta \in [0, 1]$,

$$Pred(\theta) = \theta[f(x^k) - f(z^k)] + (1 - \theta)[h(x^k) - h(y^k)]. \tag{8}$$

Choose θ_k^s, the maximum of the values of $\theta \in [0, \bar{\theta}_k]$ such that $Pred(\theta_k^s) \ge \frac{1}{2}[h(x^k) - h(y^k)]$

Step 2.3: Acceptance or rejection criterion

Define $Ared_k = \psi(x^k, \theta_k^s) - \psi(z^k, \theta_k^s)$, $Pred_k = Pred(\theta_k^s)$.

If $Ared_k \ge 0.1 Pred_k$, define $x^{k+1} = z^k$, $\theta_k = \theta_k^s$, $k \leftarrow k + 1$.

Otherwise, set $\delta = 0.5\delta$, $\bar{\theta}_k = \theta_k^s$, go to Step 2.1.

End of iteration k.

3. Inexact Restoration without derivatives

Before going further into details of the algorithm, we introduce some hypotheses and results of multivariate polynomial interpolation that we make use throughout and that can be found to a more extent in [2, Chapter 2 and 3]. The algorithm developed in this work employs linear and quadratic interpolation models to solve the subproblems of each phase, which are obtained by functional interpolation in $n + 1$ and $2n + 1$ points, respectively.

3.1. General hypotheses and basic results

From now on, we make the following assumptions in order to state theoretical and practical results.

(A1) Ω is a convex and compact set.

(A2) The Jacobian matrix of $C(x)$ satisfies the Lipschitz condition [30]:

$$\|C'(y) - C'(x)\| \le L_1\|y - x\|, \quad \text{for all } x, y \in \Omega. \tag{9}$$

(A3) The gradient of f satisfies the Lipschitz condition:

$$\|\nabla f(y) - \nabla f(x)\| \le L_2\|y - x\|, \quad \text{for all } x, y \in \Omega. \tag{10}$$

Many of the theoretical results of the algorithm developed in this work are based on the properties of linear multivariate interpolation models of the objective function and constraints.

(A4) We assume that each interpolation set $Y = \{y, z^1, \ldots, z^n\} \subset \mathbb{R}^n$, which is contained in the ball $B(y, \Delta(Y))$ of radius $\Delta(Y) = \max_{1\le i\le n} \|z^i - y\|$, is "poised" for linear interpolation, i.e., the matrix of directions $S = [z^1 - y\, z^2 - y \ldots z^n - y]$ is nonsingular.

The definition of poisedness is independent of the basis for the space of linear polynomials of degree 1. Hence, if Y is poised for the natural basis $\{1, x_1, x_2, \ldots, x_n\}$ then it is poised for any other basis chosen [2, Chapter 2].

We denote $L(x) = f(y) + gf^T(x-y)$ and $m_j^c(x) = C_j(y) + gc_j^T(x-y)$ the linear interpolating model of $f(x)$ and $C_j(x)$ on Y, respectively.

The linear model $L(x) = f(y) + gf^T(x-y)$ centered at y interpolates f at the points $y, z^1, \dots, z^n$, and gf satisfies $S^T gf = \delta(f; S)$ with $\delta(f; S) = [f(z^1) - f(y), \dots, f(z^n) - f(y)]^T$. The convex hull of a set of n+1 affinely independent points $Y = \{y, z^1, \dots, z^n\}$ is called a simplex. Since the points are affinely independent, the matrix $S = [z^1 - y \dots z^n - y]$ is nonsingular.

Given a simplex of vertices Y, the simplex gradient of f at y is defined as $S^T gf = \delta(f; S)$, which coincides with $\nabla L(y)$, the gradient of the linear model $L(x)$. Therefore, the simplex gradient of f is closely related to linear multivariate polynomial interpolation.

The geometrical properties of Y determine the quality of the corresponding $\nabla L(y)$ as an approximation to the exact gradient of the objective function.

Now, we are interested in the quality of $L(x)$ and $\nabla L(x)$ in the ball $B(y, \Delta(Y))$ with radius $\Delta(Y)$ centered at y.

Assumption (A4) gives a threshold to the difference between the functions and their interpolation models. Then, for all $x \in B(y, \Delta(Y))$, considering the scaled matrix $\bar{S} = S/\Delta(Y) = [\frac{z^1-y}{\Delta(Y)} \dots \frac{z^n-y}{\Delta(Y)}]$, we have

$$|f(x) - L(x)| \le \kappa_{ef} \Delta^2, \tag{11}$$

$$\|\nabla f(x) - \nabla L(x)\| \le \kappa_{eg} \Delta, \tag{12}$$

where $\kappa_{eg} = L_2(1 + n^{1/2}\|\bar{S}^{-1}\|/2)$ and $\kappa_{ef} = \kappa_{eg} + L_2/2$, which are given in Theorems 2.11 and 2.12 in [2].

Similarly, under the previous hypotheses the error bound between $\nabla C_j(x)$ and $\nabla m_j^c(x)$, for all $j = 1, \dots, m$, is given by $\kappa_{eg_c}\Delta = L_1(1 + n^{1/2}\|\bar{S}^{-1}\|/2)\Delta$. Then the matrix $A(y)$, an approximation of $C'(y)$, whose jth row is the transpose of $\nabla m_j^c(y)$, satisfies

$$\|C'(y) - A(y)\| \le \kappa_{e_J} \Delta, \tag{13}$$

where $\kappa_{e_J} = \sqrt{m}\kappa_{eg_c}\Delta$.

We assume that it is possible to maintain the constants κ_{ef}, κ_{eg} and κ_{e_J}, shown in the previous formulas, uniformly bounded along the iterative process of our algorithm [2, Chapter 3 and 6].

Definition 1. Given $y \in \mathbb{R}^n$ and the hyperplane $A(y)(x-y) = 0$, we define

$$\pi := \{x \in \Omega \mid A(y)(x-y) = 0\}. \tag{14}$$

Definition 2. Given gf, the simplex gradient of f at y, $y \in \mathbb{R}^n$, we also define gf_{tan} the projected direction of $-gf \in \mathbb{R}^n$ onto π as

$$gf_{tan} = P_\pi(y - gf) - y \tag{15}$$

where $P_\pi(z)$ denotes the orthogonal projection of z onto π.

The algorithm proposed in this paper is defined as follows.

Algorithm 1 (*IR-DFO*)**.** Given $x^0 \in \Omega$, $f(x^0)$, $h(x^0) = \|C(x^0)\|$, $\alpha \in (0, 1)$, $\beta > 1$, $\delta_{\min} > 0$, $\{\omega_k\} > 0$, $\sum_k^\infty \omega_k < \infty$, $\{r_k\}$, $r_k > 0$, $\lim_{k\to\infty} r_k = 0$, $\theta_{-1} \in (0, 1)$, $0 < \epsilon_M < 1$. Set $k \leftarrow 0$.

Step 0: Penalty parameter

Define $\theta_{k,-1} = \min\{1, \min\{\theta_{-1}, \dots, \theta_{k-1}\} + \omega_k\}$.

Step I: Restoration phase

If $h(x^k) = 0$, define $y^k = x^k$. Otherwise, compute $y^k \in \Omega$ solving approximately

$$\min_{y\in\Omega} h(y)^2 \quad \text{s. t. } \|y - x^k\| \le \beta h(x^k) \tag{16}$$

satisfying $h(y^k) \le \alpha h(x^k)$, by a derivative-free algorithm. If this is not possible, stop the execution of the algorithm declaring failure in improving feasibility. **END**.

Step II: Compute an approximation of $C'(y^k)$

Construct/update a set of interpolation points centered in y^k, $Y_c^k = \{y^k\} \bigcup \{z^1, z^2, \dots, z^n\}$, such that $\Delta_k = \max_{z^i \in Y_c^k}\{\|z^i - y^k\|\}$ satisfies

$$\Delta_k \le \min\{r_k, \beta_k\}, \tag{17}$$

where $\beta_k = \beta h(x^k)$ if $h(x^k) \ne 0$, else $\beta_k = r_k$.

Compute the matrix A_k, approximation of $C'(y^k)$ and define

$$\pi_k = \{z \in \Omega \mid A_k(z - y^k) = 0\}.$$

Step III: Optimization phase

III.a Choose $\delta_{k,0} \geq \delta_{\min}$. Set $i \leftarrow 0$. Obtain an approximate solution $z^{k,0}$ of (7), replacing $T(y^k)$ by π_k, such that $\|z^{k,0} - y^k\| \leq \delta_{k,0}$, satisfying $f(z^{k,0}) \leq f(y^k)$, using an algorithm without derivatives, following the scheme of Basic Algorithm 2. It must also provide an approximation of $\nabla f(y^k)$, gf, and $gf_{tan}^k = P_{\pi_k}(y^k - gf) - y^k$.

– If $z^{k,0} = y^k$, with $gf_{tan}^k = P_{\pi_k}(y^k - gf) - y^k = 0$ and $h(x^k) = h(y^k) = 0$, and $r_k < \epsilon_M$, define $x^{k+1} = z^{k,0}$. Terminate the execution of the algorithm declaring *finite convergence*.

III.b While x^{k+1} is not defined, do:

If $i > 0$, using Basic Algorithm 2, compute $z^{k,i} \in \pi_k$, such that $\|z^{k,i} - y^k\| \leq \delta_{k,i}, f(z^{k,i}) < f(y^k)$, and gf_{tan}^k.

- **Choice of penalty parameter**:

  Define $Pred_{k,i}(\theta)$ as in (8).

  Compute $\theta_{k,i}$, the maximum of the elements $\theta \in [0, \theta_{k,i-1}]$ that verifies

$$Pred_{k,i}(\theta_{k,i}) \geq \frac{1}{2}[h(x^k) - h(y^k)]. \tag{18}$$

- **Acceptance criteria of the point** $z^{k,i}$.

  Define

$$Ared_{k,i}(\theta_{k,i}) = \theta_{k,i}[f(x^k) - f(z^{k,i})] + (1 - \theta_{k,i})[h(x^k) - h(z^{k,i})]. \tag{19}$$

  – If $Ared_{k,i}(\theta_{k,i}) < 0.1\ Pred_{k,i}(\theta_{k,i})$:

  ∗ if x^k is feasible and $\delta_{k,i} \leq \min\{\epsilon_M, r_k\}$, define

$$x^{k+1} = y^k, \qquad \theta_k = \theta_{k,i}, \qquad iacc(k) = i,\ d_k = gf_{tan}^k;$$

  ∗ else, choose $\delta_{k,i+1} \in [0.1\delta_{k,i},\ 0.9\delta_{k,i}]$. Set $i \leftarrow i + 1$.

  – If $Ared_{k,i}(\theta_{k,i}) \geq 0.1\ Pred_{k,i}(\theta_{k,i})$, define

$$x^{k+1} = z^{k,i}, \qquad \theta_k = \theta_{k,i}, \qquad iacc(k) = i, \qquad d_k = gf_{tan}^k.$$

  End while.

End k-th iteration.

Remark 1. Restoration phase is defined by conditions (5) and (6) like in [19]. Notice that the objective function is not involved in such conditions.

An attractive feature of IR-type algorithms is the freedom to choose the procedure to perform the Restoration and Minimization steps. In our method, different derivative-free algorithms can be used to obtain the required decrease of infeasibility. We will give further details in Section 5.

Step III requires to solve a linearly constrained optimization problem. The basic idea of this step is the minimization of the function f subject to linear constraints and a trust-region around the restored point, controlling the feasibility achieved in Step I.

As in Martínez and Pilotta [19], in order to measure the progress to the solution, we use the merit function (2) and the same procedure for updating the penalty parameter θ.

In particular, $Ared(\theta_{k,i}) = \psi(x^k, \theta_{k,i}) - \psi(z^{k,i}, \theta_{k,i})$ measures the reduction of the merit function at the new point $z^{k,i}$ in relation to the current x^k.

When $z^{k,0} = y^k$ and the procedure continue in Step III.b, since in this case $Ared(\theta_{k,0}) = Pred(\theta_{k,0})$, after to find $\theta_{k,0}$ such that $Pred(\theta_{k,0}) \geq \frac{1}{2}[h(x^k) - h(y^k)]$, it satisfies the condition for which the k-iteration finishes defining $x^{k+1} = z^{k,0} = y^k$ and $iacc(k) = 0$.

The results below will allow us to prove that Algorithm 1 is well defined and its convergence.

Remark 2. If $x \in \mathbb{R}^n$, $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$.

Lemma 1. *Given gf, π and gf_{tan} as were defined in 1 and 2, $\epsilon > 0$. Then*

(i) $\langle gf_{tan}, gf\rangle \leq -\frac{\|gf_{tan}\|^2}{2}$

(ii) *If* $\|\nabla f(y) - gf\| < \frac{\epsilon}{2}$ *and* $\|y - P_\pi(y - \nabla f(y))\| > \epsilon$ *then* $\|gf_{tan}\| > \frac{\epsilon}{2}$.
Similarly, if $\|gf_{tan}\| > \epsilon$ *then* $\|y - P_\pi(y - \nabla f(y))\| > \frac{\epsilon}{2}$.

(iii) *If* $\|gf_{tan}\| > \epsilon$ *and* $\|\nabla f(y) - gf\| < \frac{\epsilon}{4}$ *then* $\langle \nabla f(y), gf_{tan}\rangle < 0$. *Furthermore,*

$$\langle \nabla f(y), gf_{tan}\rangle < -\frac{1}{4}\|gf_{tan}\|^2. \tag{20}$$

(iv) *If* $\|gf_{tan}\| < \frac{\epsilon}{2}$ *and* $\|\nabla f(y) - gf\| < \frac{\epsilon}{2}$ *then* $\|P_\pi(y - \nabla f(y)) - y\| < \epsilon$.

Proof. The result (i) is similar to that of [19, Section 2.6, p. 6] replacing ∇f by gf.

(ii) Since the projection P_π is non-expansive, $\|P_\pi(y-\nabla f(y)) - P_\pi(y-gf)\| \le \|\nabla f(y) - gf\|$, then it follows

$$\|y - P_\pi(y-\nabla f(y))\| \le \|y - P_\pi(y-gf)\| + \|\nabla f(y) - gf\|. \tag{21}$$

Therefore, $\|gf_{tan}\| = \|y - P_\pi(y-gf)\| \ge \|y - P_\pi(y-\nabla f(y))\| - \|\nabla f(y) - gf\| > \frac{\epsilon}{2}$, as we wanted to prove. Similarly, replacing gf with $\nabla f(y)$ in (21), it obtains the second inequality.

(iii) Since $\langle gf_{tan}, \nabla f(y)\rangle = \langle gf_{tan}, \nabla f(y) - gf\rangle + \langle gf_{tan}, gf\rangle$, then

$$\langle gf_{tan}, \nabla f(y)\rangle \le \|gf_{tan}\| \, \|\nabla f(y) - gf\| + \langle gf_{tan}, gf\rangle .$$

Considering the result of (i) we get $\langle gf_{tan}, \nabla f(y)\rangle \le \|gf_{tan}\|^2(\frac{\|\nabla f(y)-gf\|}{\|gf_{tan}\|} - \frac{1}{2})$. So, $\langle gf_{tan}, \nabla f(y)\rangle < \|gf_{tan}\|^2(\frac{1}{4} - \frac{1}{2}) = -\frac{1}{4}\|gf_{tan}\|^2$.

Therefore, under the hypotheses given, gf_{tan} is a descent direction of f.

(iv) The proof is immediate considering (21). □

Remark 3. It is known that the projected direction of $-\nabla f$ on π, if it is non-null, is a descent direction of f. The properties above give us conditions to ensure when gf_{tan} is a descent direction of the objective function f.

The following algorithm is performed to solve the minimization problem at Step III from k-th iteration of IR-DFO. In this algorithm, we follow the definitions given by Conn–Scheinberg–Vicente in [2, Chapter 2] about positive spanning set and positive basis in R^n which are:

Definition 3. The positive span of a set of vectors $[v_1, \dots, v_r]$ in R^n is the convex cone

$$\{v \in R^n : v = \alpha_1 v_1 + \cdots + \alpha_r v_r,\ \alpha_i \ge 0,\ i = 1, \dots, r\}.$$

A positive spanning set in R^n is a set of vectors whose positive span is R^n. The set $[v_1, \dots, v_r]$ is said to be positively dependent if one of the vectors is in the convex cone positively spanned by the remaining vectors, i.e., if one of the vectors is a positive combination of the others; otherwise, the set is positively independent. A positive basis in R^n is a positively independent set whose positive span is R^n.

Algorithm 2 (*Basic Algorithm for Step III (Minimization)*)**.** Given π_k, y^k, x^k, $f(y^k)$, $r_k > 0$, $\beta_k > 0$, A_k, Z_k orthogonal basis of $\mathcal{N}(A_k)$, B_k positive basis of $\mathcal{N}(A_k)$, D_k positive spanning set, $i \ge 0$, $\delta_{k,i} > 0$, $0 < \gamma_2 < 1$, $0 < \eta_1 < 1$, $tol > 0$.

Step 1. If $i = 0$, compute the simplex gradient gf, as an approximation of $\nabla f(y^k)$, by interpolation on $Y_f^k = \{y^k\} \bigcup \{z^1, z^2, \dots, z^n\}$, which is constructed or updated from a previous set, such that $\Delta(Y_f^k) \le \min\{r_k, \beta_k\}$.

Compute $gf_{tan} = P_{\pi_k}(y^k - gf) - y^k$, the projected direction of $-gf$ onto π_k.

(1.a) If $\|gf_{tan}\| \ne 0$, compute $t_{\max}^{k,0} = \min\{1, \delta_{k,0}/\|gf_{tan}\|\}$. Set $j \leftarrow 0,\ t = t_{\max}^{k,0}$.

– While $(f(y^k + t gf_{tan}) \ge f(y^k)$ and $t_{\max}^{k,0}/2^j > tol)$ do
$j \leftarrow j+1,\ t = t_{\max}^{k,0}/2^j$. End While.

– If there exists t such that $f(y^k + t gf_{tan}) < f(y^k)$, define $z^{k,0}$ such that
$f(z^{k,0}) \le \max\{f(y^k + t gf_{tan}), f(y^k) - \gamma_2,\ f(y^k) - \eta_1 \|gf_{tan}\|\delta_{k,0}\}$ and $gf_{tan}^k = gf_{tan}$. **Return**.

(1.b) Compute $\tau_{\max} = \min_{d_j \in D_k}\{1, \delta_{k,0}/\|d_j\|\}$. Set $\nu = 0$.
Find a $d_j \in D_k$ such that $f(y^k + \tau_\nu d_j) < f(y^k)$, for the greatest $\tau_\nu = \tau_{\max}/2^\nu,\ \nu = 0, 1, 2, \dots$.
If it is possible, define $\bar{z}^{k,0} = y^k + \tau_\nu d_j$, update Y_f^k and gf_{tan} and define $z^{k,0}$ such that
$f(z^{k,0}) \le \max\{f(\bar{z}^{k,0}), f(y^k) - \gamma_2,\ f(y^k) - \eta_1\|gf_{tan}\|\delta_{k,0}\}$ and $gf_{tan}^k = gf_{tan}$. **Return**.

(1.c) If it was not possible to find in item (1.a) or (1.b) a z such that $f(z) < f(y^k)$, define $z^{k,0} = y^k$, $gf_{tan}^k = gf_{tan}$. **Return**.

Step 2. If $i > 0$, compute $t_{\max}^{k,i} = \min\{1, \delta_{k,i}/\|gf_{tan}\|\}$, where $gf_{tan} = gf_{tan}^k \ne 0$.

(2.a) Find the greatest $t \in (0, t_{\max}^{k,i}]$ such that $f(y^k + t gf_{tan}) < f(y^k)$.
If there exists t with success, define $\bar{z}^{k,i} = y^k + t gf_{tan}$, go to step (2.c).
Else,

(2.b) Compute $\tau_{\max} = \min_{d_j \in D_k}\{1, \delta_{k,i}/\|d_j\|\}$.
Find the greatest $\tau \in (0, \tau_{\max}]$ such that $f(y^k + \tau d_j) < f(y^k)$ for any $d_j \in D_k$, and define $\bar{z}^{k,i} = y^k + \tau d_j$.
Update Y_f^k and gf_{tan}, using the latest evaluations of f on D_k.

(2.c) Define $z^{k,i}$, such that $f(z^{k,i}) \le \max\{f(\bar{z}^{k,i}), f(y^k) - \gamma_2,\ f(y^k) - \eta_1\|gf_{tan}\|\delta_{k,i}\}$, and $gf_{tan}^k = gf_{tan}$. **Return**.

Lemma 2. *Basic Algorithm 2 is well defined.*

Proof. When $i = 0$, after a finite number of inner iterations the procedure finishes in (1.a) and/or (1.b) with success or in (1.c) with non success.

If the projected direction of $-\nabla f(y^k)$ onto π_k is non null, although the direction gf_{tan} has not been accurately computed, the function descends on at least one direction of D_k [31]. Hence, there exists a $z^{k,0}$ such that $f(z^{k,0}) < f(y^k)$. On the other hand, when that projection is null, the procedure finishes in (1.c), after a finite number of inner iterations, accepting $z^{k,0} = y^k$.

When $i > 0$, corresponding to the case with success for $i = 0$, after a finite number of inner iterations the procedure finds a point $z^{k,i}$ such that $f(z^{k,i}) < f(y^k)$ in the trust-region with radius $\delta_{k,i} < \delta_{k,0}$. $\square$

From assumptions (A1)–(A3), Martínez and Pilotta [19] stated a bounded deterioration result for the feasibility of the point computed at the Optimization phase in relation with the point computed at the Restoration phase. More precisely, they showed that given $y \in \Omega,\ x \in T(y)$, there exists $L_1 > 0$ such that $\|C(x)\| \le \|C(y)\| + L_1\|x - y\|^2$.

Since we compute $A(y)$, the Jacobian matrix approximation of $C(y)$, we show that the deterioration of the feasibility on π_k, which depends on the radius $\Delta(Y)$ of the interpolation set, is of the order of $\Delta(Y)\|x - y\|$.

Theorem 1. *Given the system of nonlinear equations $C(x) = 0$ of the problem (1). There exists $L_1 > 0$ (independent of k) such that, if $y \in \Omega$ is computed in Restoration phase, $x \in \pi$ then*

$$\|C(x)\| \le \|C(y)\| + \kappa_{e_J}\Delta(Y)\|x - y\| + L_1\|x - y\|^2. \tag{22}$$

Proof. Since $x \in \pi,\ A(y)(x - y) = 0$. Then, the result follows from (9) and (13),

$$\|C(x)\| \le \|C(y)\| + \|C'(y) - A(y)\|\,\|(x - y)\| + L_1\|x - y\|^2 \le \|C(y)\| + \kappa_{eJ}\Delta(Y)\|x - y\| + L_1\|x - y\|^2. \quad \square$$

Remark 4. If $\Delta(Y)$ is small enough, the linear term in (22) also becomes small. Such consideration will be useful for obtaining convergence results as those obtained by Martínez and Pilotta in [19].

In the following we will show that Algorithm IR-DFO is well defined. In fact, we will prove that, when the Algorithm IR-DFO does not finished in Restoration phase with "failure in improving feasibility", there exists $\delta_{k,i}$ small enough such that $Ared_{k,i}(\theta) \ge 0.1\, Pred_{k,i}(\theta)$ and $z^{k,i}$ will be the next iterate x^{k+1} or, if x^k is feasible and $\delta_{k,i} \le \min\{\epsilon_M, r_k\}$, the iteration finishes with $x^{k+1} = x^k$.

Theorem 2. *Algorithm IR-DFO is well defined.*

Proof. Suppose Algorithm IR-DFO does not finished with "failure in improving feasibility". After some calculation we obtain that $Ared_{k,i} - 0.1Pred_{k,i} = 0.9\theta_{k,i}\left[f(x^k) - f(z^{k,i})\right] + (1-\theta_{k,i})\left[\|C(x^k)\| - \|C(z^{k,i})\|\right] - 0.1(1-\theta_{k,i})\left[\|C(x^k)\| - \|C(y^k)\|\right]$. Also, $Ared_{k,i} - 0.1Pred_{k,i}$ is equal to $0.9\theta_{k,i}\left[f(x^k) - f(z^{k,i})\right] + 0.9(1-\theta_{k,i})\left[\|C(x^k)\| - \|C(y^k)\|\right] + (1-\theta_{k,i})\left[\|C(x^k)\| - \|C(z^{k,i})\|\right] - (1-\theta_{k,i})\left[\|C(x^k)\| - \|C(y^k)\|\right]$.

Then $Ared_{k,i} - 0.1Pred_{k,i} = 0.9Pred_{k,i} + (1 - \theta_{k,i})\left[\|C(y^k)\| - \|C(z^{k,i})\|\right]$.

Hence, by (18) and $\theta_{k,i} \in [0, 1]$, we obtain

$$Ared_{k,i} - 0.1Pred_{k,i} \ge 0.45\left[\|C(x^k)\| - \|C(y^k)\|\right] - \left|\|C(y^k)\| - \|C(z^{k,i})\|\right|,$$

by (6), we have

$$Ared_{k,i} - 0.1Pred_{k,i} \ge 0.45(1 - \alpha)\|C(x^k)\| - \left|\|C(y^k)\| - \|C(z^{k,i})\|\right|.$$

If $\|C(x^k)\| \ne 0$, the first term of the right side of the last inequality is positive and, by continuity of C, the second term tends to zero as $\delta_{k,i} \to 0$. Therefore, there exists a positive $\delta_{k,i}$ such that $Ared_{k,i} \ge 0.1\, Pred_{k,i}$. This means that our algorithm is well defined when $\|C(x^k)\| \ne 0$.

If x^k is feasible, $y^k = x^k$ and $C(x^k) = C(y^k) = 0$, there are two possibilities to analyze in Step III.b. If there exists $i > 0$, such that $\delta_{k,i} \le \min\{\epsilon_M, r_k\}$ and $Ared_{k,i} - 0.1Pred_{k,i} < 0$, this step finishes with $x^{k+1} = x^k$. Otherwise, there exists $i \ge 0$ and $\delta_{k,i} > 0$ such that $Ared_{k,i} - 0.1Pred_{k,i} \ge 0$, then this step finishes with $x^{k+1} = z^{k,i}$. Hence, x^{k+1} is well defined in both cases. Consequently, Algorithm IR-DFO is well defined. $\square$

4. Convergence results of IR-DFO

4.1. Convergence to feasible points

Theorem 3. *Assume that $\{x^k\}$ is generated by Algorithm IR-DFO. Then,*

$$\lim_{k\to\infty} \psi(x^k, \theta_k) - \psi(x^{k+1}, \theta_k) = 0.$$

(i.e., $\lim_{k\to\infty} Ared_{k,iacc(k)}(\theta_{k,iacc(k)}) = 0$).

Proof. This proof is similar to that of [19, Theorem 3.4] replacing $\|C^+(x^k)\|$ by $\|C(x^k)\|$. □

Theorem 4. *If Algorithm IR-DFO does not stop at* StepI, *then*

$$\lim_{k\to\infty} \|C(x^k)\| = 0.$$

In particular, every limit point of $\{x^k\}$ *is feasible.*

Proof. This proof is similar to that of [19, Theorem 3.5] replacing $\|C^+(x^k)\|$ by $\|C(x^k)\|$. □

Remark 5. The sequence $\{y^k\}$, generated in the Restoration phase, satisfies $\lim_{k\to\infty} \|C(y^k)\| = 0$ because $\|C(y^k)\| \le \|C(x^k)\|$ and $\lim_{k\to\infty} \|C(x^k)\| = 0$ by Theorem 4. Therefore, every limit point of $\{y^k\}$ is feasible.

As a consequence of Step I of Algorithm IR-DFO, y^k satisfies $\|y^k - x^k\| \le \beta\|C(x^k)\|$ then, $\lim_{k\to\infty} \|y^k - x^k\| = 0$.

Hence, if $\bar{x}$ is a limit point of $\{x^k\}$, also $\bar{x}$ is a limit point of $\{y^k\}$ since $\|y^k - \bar{x}\| \le \|y^k - x^k\| + \|x^k - \bar{x}\|$. Analogously, every limit point of $\{y^k\}$ is a limit point of $\{x^k\}$.

4.2. Convergence to optimality

If the used interpolation models of the objective function are good enough, $\|gf^k_{tan}\|$ is an indicator of optimality as we have shown in Lemma 1. We will prove it cannot be bounded away from zero when x^k is almost feasible in the sense of the previous section.

We will prove the convergence of IR-DFO proceeding by contradiction, as Martínez and Pilotta in [19]. So, we will assume $\|gf^k_{tan}\|$ is bounded away from zero, assuming that the used interpolation models are good enough, for k large enough. Then we will show that the results obtained from this hypothesis lead us to a contradiction.

Notation. For simplicity, we rename gf^k_{tan} by d_k.

Hypothesis B

There exist $\epsilon > 0$ and $k_0 \in \mathbb{N}$ such that

$$\|d_k\| \ge \epsilon \quad \text{for all } k \ge k_0.$$

Theorem 5. *Suppose that Hypothesis B holds. Then, there exist* $k_1 \ge k_0$ *and* $c_2,\ c_3 > 0$ *(independent of* k*) such that, whenever* y^k *is defined and* $z^{k,i}$ *is computed by* Algorithm 2, *we have that*

$$f(z^{k,i}) \le f(y^k) - \min\{c_2\|d_k\|^2, c_3\delta_{k,i}\|d_k\|\}, \quad \textit{for all } k \ge k_1.$$

Proof. By (10), $y^k + d_k \in \Omega$, for all $t \in [0, 1]$,

$$f(y^k + td_k) \le f(y^k) + 0.1t\left\langle\nabla f(y^k), d_k\right\rangle + 0.9t\left\langle\nabla f(y^k), d_k\right\rangle + \frac{t^2L_2}{2}\|d_k\|^2.$$

Since $r_k \downarrow 0$, there exists $k_1 \ge k_0$ such that for all $k \ge k_1,\ r_k < \frac{\epsilon}{4L_2\kappa_{eg}}$. Then, by (20), for all $k \ge k_1$

$$f(y^k + td_k) \le f(y^k) + 0.1t\left\langle\nabla f(y^k), d_k\right\rangle + t\|d_k\|^2\left(-\frac{0.9}{4} + \frac{tL_2}{2}\right).$$

Furthermore, considering again (20) and $t \le \frac{0.9}{2L_2}$ we have that

$$f(y^k + td_k) \le f(y^k) - \frac{0.1t}{4}\|d_k\|^2.$$

As $t\|d_k\| \le \delta_{k,i}$, then for $t \le \min\{1, \frac{\delta_{k,i}}{\|d_k\|}, \frac{0.9}{2L_2}\}$ we have that $f(y^k + td_k) \le f(y^k) - \frac{0.1t}{4}\|d_k\|^2$. Hence,

$$f(y^k + td_k) \le f(y^k) - \frac{0.1}{4}\min\left\{1, \frac{\delta_{k,i}}{\|d_k\|}, \frac{0.9}{2L_2}\right\}\|d_k\|^2.$$

Thus, $f(y^k + td_k) \le f(y^k) - \min\left\{\frac{0.1}{4}\|d_k\|^2, \frac{0.1}{4}\delta_{k,i}\|d_k\|, \frac{0.1}{4}\frac{0.9}{2L_2}\|d_k\|^2\right\}$. Therefore, defining $c_3 = \frac{0.1}{4}$ and $c_2 = c_3\min\left\{1, \frac{0.9}{2L_2}\right\}$, we obtain

$$f(y^k + td_k) \le f(y^k) - \min\left\{c_2\|d_k\|^2, c_3\delta_{k,i}\|d_k\|\right\}.$$

So, for all $k \ge k_1,\ f(z^{k,i}) \le f(y^k) - \min\left\{c_2\|d_k\|^2, c_3\delta_{k,i}\|d_k\|\right\}$. □

Lemma 3. *Suppose that Hypothesis B holds. There exists $k_2,\ k_2 \geq k_1$, such that if x^k is feasible ($C(x^k) = 0$) for some $k \geq k_2$ then there exist $i \geq 0$ and $\delta_{k,i} \geq \min\{r_k, \epsilon_M\}$ such that*

$$Ared_{k,i} - 0.1Pred_{k,i} \geq 0 \tag{23}$$

is satisfied in StepIII.b *of Algorithm IR-DFO.*

Proof. If x^k is feasible, $\|C(x^k)\| = 0$ and $y^k = x^k$. If $z^{k,i},\ i \geq 0$, is such that $f(z^{k,i}) \leq f(y^k)$,

$$Ared_{k,i} - 0.1Pred_{k,i} = 0.9\theta_{k,i}\big[f(x^k) - f(z^{k,i})\big] - (1 - \theta_{k,i})\|C(z^{k,i})\|.$$

Since $f(x^k) = f(y^k)$, for all $i,\ i \geq 0$ and $\theta \in (0, 1]$, it follows
$Pred_{k,i}(\theta) = \theta(f(x^k) - f(z^{k,i})) \geq \frac{1}{2}\big(\|C(x^k)\| - \|C(y^k)\|\big) = 0$. Therefore, $\theta_{k,i} = \theta_{k,-1}$ for all $i,\ i \geq 0$. Hence,

$$Ared_{k,i} - 0.1Pred_{k,i} = 0.9\theta_{k,-1}[f(x^k) - f(z^{k,i})] - (1 - \theta_{k,-1})\|C(z^{k,i})\|.$$

Suppose that for all $k \geq k_1$, if $C(x^k) = 0$ then $Ared_{k,i} - 0.1Pred_{k,i} \geq 0$ does not hold for $0 \leq i \leq i_0^k - 1$, being $\delta_{k,i} \geq \min\{r_k, \epsilon_M\}$ and i_0^k the first index which satisfies $\delta_{k,i_0^k} < \min\{r_k, \epsilon_M\}$.

Then, denoting $i_p^k = i_0^k - 1$, by Theorems 1 and 5, for $k \geq k_1$ we get $0 > Ared_{k,i_p^k} - 0.1Pred_{k,i_p^k} \geq 0.9\theta_{k,-1}\min\{c_2\|d_k\|^2, c_3\|d_k\|\delta_{k,i_p^k}\} - (1 - \theta_{k,-1})(\kappa_{e_J}\Delta_k\delta_{k,i_p^k} + L_1\delta^2_{k,i_p^k})$. Therefore,

$$0.9\theta_{k,-1}\|d_k\|\min\{c_2\|d_k\|,\ c_3\delta_{k,i_p^k}\} < (1 - \theta_{k,-1})\left(\kappa_{e_J}\Delta_k\delta_{k,i_p^k} + L_1\delta^2_{k,i_p^k}\right).$$

Hence, since $0 < \delta_{k,i_0^k} < \delta_{k,i_p^k} \leq 10\delta_{k,i_0^k}$ by the condition of the decrease at Step III.b,

$$0.9\theta_{k,-1}\|d_k\|\min\{c_2\|d_k\|/\delta_{k,i_0^k},\ c_3\} < (1 - \theta_{k,-1})(10\kappa_{e_J}\Delta_k + 100L_1\delta_{k,i_0^k}). \tag{24}$$

If $\|d_k\| < \delta_{k,i_0^k}$, then $\|d_k\| < r_k$ ($\delta_{k,i_0^k} < r_k$). Otherwise, if $\|d_k\| \geq \delta_{k,i_0^k}$, since $\delta_{k,i_0^k} < r_k$, $\Delta_k \leq r_k$ and $\frac{\|d_k\|}{\delta_{k,i_0^k}} \geq 1$, in (24) we have that

$$0.9\theta_{k,-1}\|d_k\|\min\{c_2,\ c_3\} < (1 - \theta_{k,-1})(10\kappa_{e_J}r_k + 100L_1r_k).$$

Then, considering $A = \min\{c_2,\ c_3\}$ and $B = \frac{1-\theta_{k,-1}}{0.9\theta_{k,-1}}10(\kappa_{e_J} + 10L_1)$, it obtains $\|d_k\| < \frac{B}{A}r_k$. Therefore, for all $k \geq k_1$ such that $C(x^k) = 0$, it obtains $\|d_k\| < r_k$ or $\|d_k\| < \frac{B}{A}r_k$. It would have that $\|d_k\|$ tends to zero because $r_k \downarrow 0$ and this contradicts Hypothesis B.

Hence, if Hypothesis B holds, there exists $k_2 \geq k_1$ such that if $k \geq k_2$ and $C(x^k) = 0$, there exists $i \geq 0$ such that $\delta_{k,i} \geq \min\{r_k, \epsilon_M\}$ and $Ared_{k,i} - 0.1Pred_{k,i} \geq 0$. $\square$

Lemma 4. *Suppose that Hypothesis B holds. Then, there exists $c_4, c_5 > 0$ (independent of k) such that*

$$f(y^k) - f(z^{k,i}) \geq \min\{c_4, c_5\delta_{k,i}\}$$

for all $k \geq k_2,\ i = 0, 1, \ldots, iacc(k)$.

Proof. The result follows trivially from Theorem 5 and Hypothesis B. $\square$

Lemma 5. *Suppose that Hypothesis B holds. Then, there exist $\gamma_1, \gamma_2 > 0$, independent of k, and $i \geq 0$ such that $\|C(x^k)\| \leq \min\{\gamma_2, \gamma_1\delta_{k,i}\}$ implies that $\theta_{k,i} = \theta_{k,i-1}$, for all $k \geq k_2$.*

Proof. This proof is similar to that of [19, Lemma 4.2] replacing $\|C^+(x^k)\|$ by $\|C(x^k)\|$. $\square$

Lemma 6. *Suppose that Hypothesis B holds. If $\|C(x^k)\|$ is sufficiently small ($C(x^k) \neq 0$), a step $\delta_{k,i}$ that satisfies*

$$\frac{\gamma_1}{10}\delta_{k,i} \leq \|C(x^k)\|, \tag{25}$$

is necessarily accepted in StepIII.b *of IR-DFO, where γ_1 is defined in Lemma 5.*

Proof. If (25) holds then, by (6) and (18)

$$Pred_{k,i} \geq \frac{1}{2}\left[\|C(x^k)\| - \|C(y^k)\|\right] \geq \frac{1-\alpha}{2}\|C(x^k)\| \geq \frac{(1-\alpha)\gamma_1}{20}\delta_{k,i}.$$

So, (25) implies that

$$\delta_{k,i} \leq \frac{20}{(1-\alpha)\gamma_1} Pred_{k,i}. \tag{26}$$

By Theorem 1 and $1-\theta_{k,i} > -1$,
$Ared_{k,i} = Pred_{k,i} + (1-\theta_{k,i})\left[\|C(y^k)\| - \|C(z^{k,i})\|\right] \geq Pred_{k,i} - \kappa_{e_J}\Delta_k\delta_{k,i} - L_1\delta_{k,i}^2$. Then, by (26), it obtains

$$Ared_{k,i} \geq Pred_{k,i} - \kappa_{e_J}\Delta_k \frac{20}{(1-\alpha)\gamma_1} Pred_{k,i} - L_1\delta_{k,i}\frac{20}{(1-\alpha)\gamma_1} Pred_{k,i},$$

and (25) implies that $Ared_{k,i} \geq Pred_{k,i} - \kappa_{e_J}\Delta_k \frac{20}{(1-\alpha)\gamma_1} Pred_{k,i} - L_1 \frac{10}{\gamma_1}\|C(x^k)\| \frac{20}{(1-\alpha)\gamma_1} Pred_{k,i}$. The term of the right side is equal to $Pred_{k,i}\left[1 - \kappa_{e_J}\Delta_k \frac{20}{(1-\alpha)\gamma_1} - L_1\frac{10}{\gamma_1}\|C(x^k)\|\frac{20}{(1-\alpha)\gamma_1}\right]$. Then, considering that $\Delta_k \leq \beta\|C(x^k)\|$, that term is greater than or equal to

$$Pred_{k,i}\left[1 - \kappa_{e_J}\beta\|C(x^k)\|\frac{20}{(1-\alpha)\gamma_1} - L_1\frac{10}{\gamma_1}\|C(x^k)\|\frac{20}{(1-\alpha)\gamma_1}\right].$$

Hence, it obtains

$$Ared_{k,i} \geq Pred_{k,i}\left[1 - \|C(x^k)\|\left(\frac{\kappa_{e_J}\beta\gamma_1 20 + L_1 200}{(1-\alpha)\gamma_1^2}\right)\right].$$

Then, if (25) holds and $\|C(x)\| \leq \frac{0.9(1-\alpha)\gamma_1^2}{\kappa_{e_J}\gamma_1\beta 20 + 200L_1} = H$, the trial point $z^{k,i}$ is accepted in Step III.b of Algorithm IR-DFO. □

Lemma 7. *Suppose that Hypothesis B holds. Then, there exists $\bar{\theta} > 0$ such that $\theta_k \geq \bar{\theta}$ for all $k \geq k_3$, with $k_3 \geq k_2$.*

Proof. Let $\epsilon_2 = \min\{\gamma_2, H, \gamma_1\delta_{\min}\}$, where γ_1 and γ_2 are defined in Lemma 5 and $H = \frac{0.9(1-\alpha)\gamma_1^2}{\kappa_{e_J}\gamma_1\beta 20 + 200L_1}$ is defined in Lemma 6.

Let $k_3 \geq k_2 \geq k_1$ be such that $\|C(x^k)\| \leq \epsilon_2$ for all $k \geq k_3$. Since $\delta_{\min} \geq \frac{\|C(x^k)\|}{\gamma_1}$, this implies that, for all $k \geq k_3$, $\delta_{k,0} \geq \frac{\|C(x^k)\|}{\gamma_1}$, because $\frac{\epsilon_2}{\gamma_1} \leq \delta_{\min} \leq \delta_{k,0}$.

Therefore, a possible trust-region radius such that $\delta_{k,i} < \frac{\|C(x^k)\|}{\gamma_1}$ cannot correspond to $i = 0$. Hence, it is preceded by a $\delta_{k,i-1}$, which necessarily verifies $\delta_{k,i-1} \leq 10\delta_{k,i} < 10\frac{\|C(x^k)\|}{\gamma_1}$. This means that $\delta_{k,i-1}$ satisfies (25) and by Lemma 6 the corresponding point $z^{k,i-1}$ would be accepted for all $k \geq k_3$.

Therefore, $\delta_{k,i}$ must satisfy $\delta_{k,i} \geq \frac{\|C(x^k)\|}{\gamma_1}$. Then, since $\|C(x^k)\| \leq \epsilon_2$ for all $k \geq k_3$, and $\|C(x^k)\| \leq \delta_{k,i}\gamma_1$, by Lemma 5, the penalty parameter $\theta_{k,i}$ is never decreased for all $k \geq k_3$, $i = 0, 1, \ldots, iacc(k)$. Hence, there exists $\bar{\theta} > 0$ such that $\theta_k \geq \bar{\theta}$ for all $k \geq k_3$, $k \in \{1, 2, \ldots, \}$ and this implies the desired result. □

Finally, in the next Theorem 6, we prove that Hypothesis B is false.

Theorem 6. *Let $\{x^k\}$ be an infinite sequence generated by Algorithm IR-DFO. Then, there exists K_2, an infinite subset of $\mathbb{N}$, such that*

$$\lim_{\substack{k\to\infty \\ k\in K_2}} \|d_k\| = 0. \tag{27}$$

Proof. Suppose that the thesis of the theorem is not true. Then, given $\epsilon > 0$, there exists $k_0 \in \mathbb{N}$ such that Hypothesis B is true.

By definition of $Ared_{k,i} - 0.1Pred_{k,i}$ and Theorem 1, $Ared_{k,i} - 0.1Pred_{k,i} = 0.9\{\theta_{k,i}[f(x^k) - f(z^{k,i})] + (1-\theta_{k,i})[\|C(x^k)\| - \|C(y^k)\|]\} + (1-\theta_{k,i})[\|C(y^k)\| - \|C(z^{k,i})\|] \geq 0.9\theta_{k,i}[f(y^k) - f(z^{k,i})] + 0.9\theta_{k,i}[f(x^k) - f(y^k)] - (1-\alpha)|C(x^k)| - \kappa_{e_J}\Delta_k\delta_{k,i} - L_1\delta_{k,i}^2$.
Then, by Lemma 4, there exists $c_4, c_5 > 0$ such that

$$f(y^k) - f(z^{k,i}) \geq \min\{c_4, c_5\delta_{k,i}\}, \quad \text{for all } k \geq k_1,\ k_1 \geq k_0,\ i = 0, 1, \ldots, iacc(k).$$

By Lemma 7, (5) and $\Delta_k \leq \beta\|C(x^k)\|$,

$$\delta_{k,i} \geq \frac{\|C(x^k)\|}{\gamma_1} \geq \frac{\Delta_k}{\beta\gamma_1} \quad \text{for all } k \geq k_3 \geq k_2 \geq k_1,\ i = 0, 1, \ldots, iacc(k).$$

Therefore, $Ared_{k,i} - 0.1Pred_{k,i} \geq 0.9\bar{\theta}\min\{c_4, c_5\delta_{k,i}\} - c\beta\|C(x^k)\| - c_6\delta_{k,i}^2$, for all $k \geq k_3 \geq k_2 \geq k_1,\ i = 0, 1, \ldots, iacc(k)$, where c is a constant that depends on a bound of $\|\nabla f(x)\|$ on Ω and $c_6 = \kappa_{e_J}\beta\gamma_1 + L_1$.

Let us define $\bar{\delta} = \min\left\{\left(\frac{0.45\bar{\theta}c_4}{c_6}\right)^{\frac{1}{2}}, \frac{0.45\bar{\theta}c_5}{c_6}\right\}$. If $\delta_{k,i} \leq \bar{\delta}$ then

$$c_6\delta_{k,i}^2 \leq 0.45\bar{\theta}\min\left\{c_4, c_5\delta_{k,i}\right\}, \tag{28}$$

so, when $\delta_{k,i} \leq \bar{\delta}$ we have that

$$Ared_{k,i} - 0.1Pred_{k,i} \geq 0.45\bar{\theta}\min\{c_4, c_5\delta_{k,i}\} - c\beta\|C(x^k)\|, \tag{29}$$

for all $k \geq k_3,\ i = 0, 1, \ldots, iacc(k)$. Let $k_4 \geq k_3$ such that

$$c\beta\|C(x^k)\| \leq 0.45\bar{\theta}\min\left\{c_4, c_5\frac{\bar{\delta}}{10}\right\}, \tag{30}$$

for all $k \geq k_4$. By (29) and (30) we have that for all $k \geq k_4$, if $i \in \mathbb{N}$ correspond to the first trust-region radius $\delta_{k,i}$ less than or equal to $\bar{\delta}$ (so, $\bar{\delta} \geq \delta_{k,i} \geq \frac{\bar{\delta}}{10}$),

$$Ared_{k,i} - 0.1Pred_{k,i} \geq 0. \tag{31}$$

This means that $\delta_{k,i} \geq \frac{\bar{\delta}}{10}$ must be accepted. Then,

$$\delta_{k,iacc(k)} \geq \frac{\bar{\delta}}{10}, \quad \text{for all } k \geq k_4. \tag{32}$$

Notice that, $f(y^k) - f(x^k) \geq -\left|f(y^k) - f(x^k)\right|$ and $(1 - \theta_{k,iacc(k)})(1 - \alpha) > -1$ by (6), then

$$(1 - \theta_{k,iacc(k)})\left[\|C(x^k)\| - \|C(y^k)\|\right] \geq (1 - \theta_{k,iacc(k)})(1 - \alpha)\|C(x^k)\| \geq -\|C(x^k)\|.$$

Then, if $k \geq k_4$, by Lemma 4, Lemma 7 and (5), considering

$$Pred_{k,iacc(k)} = \theta_{k,iacc(k)}\left[f(x^k) - f(z^{k,i})\right] + (1 - \theta_{k,iacc(k)})\left[\|C(x^k)\| - \|C(y^k)\|\right],$$

which is equal to $\theta_{k,iacc(k)}\left[f(y^k) - f(z^{k,i})\right] + \theta_{k,iacc(k)}\left[f(y^k) - f(x^k)\right] + (1 - \theta_{k,iacc(k)})\left[\|C(x^k)\| - \|C(y^k)\|\right]$, it follows $Pred_{k,iacc(k)} \geq \bar{\theta}\left[f(y^k) - f(z^{k,i})\right] - \left|f(y^k) - f(x^k)\right| - \|C(x^k)\|$, and

$$Pred_{k,iacc(k)} \geq \bar{\theta}\min\left\{c_4, \frac{c_5\bar{\delta}}{10}\right\} - c_7\|C(x^k)\|, \tag{33}$$

where c_7 is a constant that depends on the bound of $\|\nabla f(x)\|$ on Ω and β of $\|y^k - x^k\| \leq \beta\|C(x^k)\|$.

Let $k_5 \geq k_4$ such that $c'\|C(x^k)\| \leq 0.5\bar{\theta}\min\left\{c_4, \frac{c_5\bar{\delta}}{10}\right\}$, for all $k \geq k_5$.

Then, we obtain $Pred_{k,iacc(k)} \geq 0.5\bar{\theta}\min\{c_4, \frac{c_5\bar{\delta}}{10}\}$, for all $k \geq k_5$. This and (31) imply that $Ared_{k,iacc(k)}$ is bounded away from zero for all $k \geq k_5$ and we have a contradiction with the result of Theorem 3. Hence, Hypothesis B cannot be true, as we wanted to prove. □

Theorem 7. *Let $\{y^k\}$ be the infinite sequence generated by Algorithm IR-DFO, with y^k the solution found in Restoration phase. Then,*

$$\lim_{k\in K_2}\|P_{\pi_k}(y^k - \nabla f(y^k)) - y^k\| = 0. \tag{34}$$

Proof. By (12), $\{\Delta_k\} \downarrow 0$, the result of Theorem 6 and (21)

$$\|y^k - P_{\pi_k}(y^k - \nabla f(y^k))\| \leq \|y^k - P_{\pi_k}(y^k - gf)\| + \|\nabla f(y^k) - gf\|,$$

it obtains $\lim_{k\in K_2}\|P_{\pi_k}(y^k - \nabla f(y^k)) - y^k\| = 0$, as we wanted to prove. □

Theorem 8. *Assume that $\{x^k\}$ is generated by Algorithm IR-DFO and Assumptions* A1–A4 *hold. Let $\{y^k\}$ be, $k \in K_2$, like in Theorem 7, and $\bar{x}$ the limit point of this sequence.*

If $\lim_{k\to\infty}\|A_k - C'(y^k)\| = 0$ and the Mangasarian–Fromovitz constraint qualification (MFCQ) holds at $\bar{x}$, then $\bar{x}$ fulfills the Karush–Kuhn–Tucker (KKT) conditions [30].

Proof. The optimal solution of the linear constrained problem

$$\min \|p-(y^k-\nabla f(y^k))\|^2,$$
$$\text{s.t. } p\in\pi_k=\{p\in\Omega \mid A_k(p-y^k)=0\}$$

is $p_k=P_{\pi_k}(y^k-\nabla f(y^k))$. Hence, p_k satisfies the optimality condition for this problem.

If $\Omega=\{x\in\mathbb{R}^n : L_i\le x_i\le U_i\}$, there exists $\lambda_k\in\mathbb{R}^m$, $\mu_k^u\in\mathbb{R}^n_+$, and $\mu_k^l\in\mathbb{R}^n_+$ such that

$$p_k-y^k=(-\nabla f(y^k))-A_k^T\lambda_k-\sum_{i=1}^n\mu_{k,i}^u e_i-\sum_{i=1}^n\mu_{k,i}^l(-e_i),$$

with $A_k(p_k-y^k)=0$, and $\mu_{k,i}^u\ge 0$, $\mu_{k,i}^l\ge 0$, $\mu_{k,i}^u(U_i-p_{k,i})=0$, and $\mu_{k,i}^l(p_{k,i}-L_i)=0$, $i=1,\dots,n$.

Since $\|p_k-y^k\|$ tends to zero, by Theorem 7, also $\bar{x}$ is a feasible limit point of $\{p_k\}$.

Set $I_U=\{i\in\{1,2,\dots,n\}:\bar{x}_i=U_i\}$, $I_L=\{i\in\{1,2,\dots,n\}:\bar{x}_i=L_i\}$ and $I_0=\{i\in\{1,2,\dots,n\}:L_i<\bar{x}_i<U_i\}$.

Hence, there exists $k_0\in K_2$ such that for $k>k_0$, $k\in K_2$, p_k satisfies $L_i<p_{k,i}<U_i$, for $i\in I_0$ (because $p_k\to\bar{x}$), then $\mu_{k,i}^u=0$ for all $i\in I_0$.

Consequently, for $k>k_0$, $k\in K_2$ the optimality condition implies

$$p_k-y^k=-\nabla f(y^k)-A_k^T\lambda_k-\sum_{i\in I_U}\mu_{k,i}^u e_i-\sum_{i\in I_L}\mu_{k,i}^l(-e_i), \tag{35}$$

$A_k(p_k-y^k)=0$, and $\mu_{k,i}^u\ge 0$, $\mu_{k,i}^l\ge 0$, $\mu_{k,i}^u(U_i-p_{k,i})=0$, $i\in I_U$ and $\mu_{k,i}^l(p_{k,i}-L_i)=0$, $i\in I_L$.

By Carathéodory's Theorem (convex hull) [30, (Exercise B.1.7, p. 689)], for all $k\in K_2$, $k>k_0$ there exist $I^k\subseteq\{1,2,\dots,n\}$, $I_U^k\subseteq I_U$, and $I_L^k\subseteq I_L$ such that $\left\{\{a_i^k\}_{i\in I^k},\ \{e_i\}_{i\in I_U^k},\{-e_i\}_{i\in I_L^k}\right\}$ is linearly independent. Therefore, denoting $a_i^k=[A_k^T]_i$, the ith column of A_k^T,

$$p_k-y^k=(-\nabla f(y^k))-\sum_{i\in I^k}\bar{\lambda}_{k,i}a_i^k-\sum_{i\in I_U^k}\bar{\mu}_{k,i}^u e_i-\sum_{i\in I_L^k}\bar{\mu}_{k,i}^l(-e_i).$$

Since there is only a finite number of possible index sets, there exists an infinite set $K_3\subset\{k\in K_2: k>k_0\}$, such that the sets I_k, I_U^k, I_L^k are repeated. So, for $k\in K_3$, $I^k=\bar{I}$, $I_U^k=\bar{I}_U$, $I_L^k=\bar{I}_L$. Then, for all $k\in K_3$,

$$p_k-y^k=(-\nabla f(y^k))-\sum_{i\in\bar{I}}a_i^k\bar{\lambda}_{k,i}-\sum_{i\in\bar{I}_U}\bar{\mu}_{k,i}^u e_i-\sum_{i\in\bar{I}_L}\bar{\mu}_{k,i}^l(-e_i), \tag{36}$$

and the vectors $\{\{a_i^k\}_{i\in\bar{I}},\{e_i\}_{i\in\bar{I}_U},\{-e_i\}_{i\in\bar{I}_L}\}$ are linearly independent.

Let $S_k=\max\left\{\max\{|\bar{\lambda}_{k,i}|,\ i\in\bar{I}\},\max\{\bar{\mu}_{k,i}^u,\ i\in\bar{I}_U\},\max\{\bar{\mu}_{k,i}^l, i\in\bar{I}_L\}\right\}$, $k\in K_3$.

If $\{S_k\}_{k\in K_3}$ is unbounded, dividing both sides of (36) by S_k, $\frac{p_k-y^k}{S_k}=$

$$\frac{-\nabla f(y^k)}{S_k}-\sum_{i\in\bar{I}}(a_i^k-\nabla C_i(y^k)+\nabla C_i(y^k))\frac{\bar{\lambda}_{k,i}}{S_k}-\sum_{i\in\bar{I}_U}\frac{\bar{\mu}_{k,i}^u}{S_k}e_i-\sum_{i\in\bar{I}_L}\frac{\bar{\mu}_{k,i}^l}{S_k}(-e_i), \tag{37}$$

$\frac{\bar{\mu}_{k,i}^u}{S_k}\ge 0$ and $\frac{\bar{\mu}_{k,i}^l}{S_k}\ge 0$. As for each k there is a coefficient equal to 1 or -1, there exists an infinite set $K_4\subset K_3$ for which the indices of the coefficients equal to 1(or -1) coincide. Taking limits on both sides in (37), for $k\in K_4$, we obtain that

$$0=-\sum_{i\in\bar{I}}\nabla C_i(\bar{x})\bar{\lambda}_i-\sum_{i\in\bar{I}_U}\bar{\mu}_i^u e_i-\sum_{i\in\bar{I}_L}\bar{\mu}_i^l(-e_i),$$

$\bar{\mu}_i^u\ge 0$ and $\bar{\mu}_i^l\ge 0$, with some coefficients non null, because for each k there is a coefficient equal to 1 (or -1). Then, that result implies the linear dependence of the vectors involved. Hence, the previous result contradicts our hypotheses, because in $\bar{x}$ the MFCQ constraint qualification holds. Therefore, S_k must be bounded.

Consequently, since $\{S_k\}_{k\in K_3}$ is bounded, $\lim_{k\in K_3}\bar{\lambda}_{k,i}=\lambda_i$, $\lim_{k\in K_3}\bar{\mu}_{k,i}^u=\mu_i^u$, and $\lim_{k\in K_3}\bar{\mu}_{k,i}^l=\mu_i^l$. Then, taking limits on both sides in (36), it obtains

$$0=(-\nabla f(\bar{x}))-\sum_{i\in\bar{I}}\nabla C_i(\bar{x})\lambda_i-\sum_{i\in\bar{I}_U}\mu_i^u e_i-\sum_{i\in\bar{I}_L}\mu_i^l(-e_i),$$

$\mu_i^u\ge 0$, $\mu_i^l\ge 0$, $\mu_i^u(U_i-\bar{x}_i)=0$, for all $i\in\bar{I}_U$ and $\mu_i^l(\bar{x}_i-L_i)=0$, $i\in\bar{I}_L$. Therefore, $\bar{x}$ satisfies the first order KKT conditions. $\square$

5. Numerical experiments

In this section we present some computational results obtained with two Fortran 77 implementations of IR-DFO algorithm. These experiments were run on a personal computer with INTEL(R) Core (TM) 2 Duo CPU E8400 at 3.00 GHz and 3.23 GB of RAM. As it is usual in derivative-free optimization articles we are interested in the number of function evaluations needed for satisfying the stopping criteria.

Our results are compared to those obtained with the IR method of Bueno et al. in [28], and to those obtained with Powell's software COBYLA [32], a trust-region method for constrained problems that models the objective and constraint functions by linear interpolation. The cited IR method, which is also a method based on the Inexact Restoration framework, solves nonlinear problems in which the derivatives of the objective function are not available, whereas the derivatives of the constraints are.

5.1. Details on the implementation of IR-DFO algorithm

We have considered two versions of IR-DFO, IR-DFOBQA and IR-DFOTRB. The only difference between them is how the Restoration phase is solved. In IR-DFOBQA we used BOBYQA [9], while in IR-DFOTRB we used TRB-Powell [4]. Both derivative-free methods solve bound constrained optimization problems. They are based on quadratic approximations of the objective function and trust region techniques.

We used in both versions of IR-DFO the following general parameters: $\alpha = 0.7$, $\beta = 100$, $r_k = (m+1)/(2+k)^2$, $\omega_k = (m+1)/(1+k)^2$, where m is the number of equality constraints of problems, $\epsilon_M = 10^{-5}$, $\delta_{\min} = 0.5$.

Given an iterate x^k of IR-DFO, at the Restoration phase we apply BOBYQA (or TRB-Powell), starting from the initial point $u^0 = x^k$. It iterates until finding a new iterate u^j, such that satisfies suitable conditions for defining $y^k = u^j$, i.e., satisfying the descent condition $h(y^k) \leq \alpha h(x^k)$ and $\|y^k - x^k\| \leq \beta h(x^k)$, for fixed parameters $0 < \alpha < 1$ and $\beta > 0$.

Both iterative algorithms generate a sequence $\{u^j\}$, for $j = 0, 1, 2, \ldots$, with $u^j \in \Omega_k$, being $\Omega_k = \{y \in \Omega : \|y - x^k\|_\infty \leq \frac{\beta}{\sqrt{n}} h(x^k)\}$. We used $\rho_{beg} = \min\{0.5, \frac{\beta}{\sqrt{n}} \|C(x^k)\|\}$, for the initial radius of the trust-region, and the other parameters were the default parameters of both methods [4,9].

The Optimization phase consists of minimizing the objective function, inside a trust-region, with linear constraints. We solved this problem, without using derivatives, with an implemented algorithm ad-hoc following the scheme 2. This algorithm solves approximately the minimization of a linear model of the objective function in a set which approximates the feasible region.

In this implementation of the Optimization phase we used the DLSVRR and DQPROG subroutines of the IMSL Fortran Numerical Library (Visual Fortran). The first computes the singular value decomposition (USV) of the matrix A_k and the projection of gf onto $N(A_k)$. The second performs the projection of gf on the approximate feasible set when the variables are bounded.

Step II requires the calculation of the simplex gradients of C_j, for $j = 1, \ldots, m$, which requires to select a set of interpolation points. In the first iteration we construct the set $Y_c^0 = \{y^0, z^1, \ldots, z^n\}$ for computation the models $m_j^c(x) = C_j(y^0) + gc_j^T(x - y^0)$, $j = 1, \ldots, m$, and generating the matrix $A(y^0) = A_0$, an approximation of $C'(y^0)$. We consider $z^i - y^0 = \rho_0 e_i$ and the corresponding values $C_j(z^i)$, for $i = 1, \ldots, n$ and $j = 1, \ldots, m$, $\rho_0 < r_0$.

Step III also requires to compute the model $L(x) = f(y^k) + gf^T(x - y^k)$. In the first iteration, we used the vectors of the matrix V of the decomposition USV of A_0 to obtain the model $L(x) = f(y^0) + gf^T(x - y^0)$, considering the set $Y_f^0 = \{y^0, z^1, \ldots, z^n\}$, where $z^i = y^0 + \rho_0 v_i$ and $f(z^i)$, for $i = 1, \ldots, n$. If there are not active bound constraints at y^0 it is possible to obtain gf using only the vectors of a basis of $N(A(y^0))$ extracted from V.

In the following iterations Y_c^k and Y_f^k are updated, adding the new y^k as the center of them and eliminating a point z_t, the farthest from the center, trying to maintain the linear independence of the directions.

In some iterations of this preliminary implementation the interpolation sets are newly constructed, while in others they are updated from the previous ones. The construction takes place in the first iteration and whenever it is not possible to preserve the linear independence of the directions easily.

In this implementation we have not constructed D_k, the positive spanning set of the active constraints at y^k. As noted in Remark 1, if $z^{k,0} = y^k$ because f does decrease for $i = 0$, Step III finishes with $x^{k+1} = y^k$. Neither has it been taken advantage of the information that comes from the last model calculated by the method that solves the Restoration phase (Step I).

Both shortcomings in the current implementation will be the subject of a careful study in a future implementation which could result in a significant decrease of the number of functional evaluations.

5.2. Test problems

We have selected a set of 32 nonlinear programming problems defined by Hock and Schittkowski [33] with nonlinear constraints and/or box constraints. The dimension of these problems (n) varies from 2 to 10 and the number of nonlinear constraints (m) varies from 1 to 6. The selected test set contains some of the problems used by the authors of [28] in their numerical tests.

Table 1
Characteristics of selected problems.

Problem	n	m	nL	nU	$nL + nU$	f^*
6	2	1	0	0	0	0.0000D+00
7	2	1	0	0	0	−1.7321D+00
8	2	2	0	0	0	−1.0000D+00
9	2	1	0	0	0	−5.0000D−01
14	3	2	1	0	0	1.3935D+00
18	4	2	2	0	2	5.0000D+00
26	3	1	0	0	0	0.0000D+00
27	3	1	0	0	0	4.0000D−02
32	4	2	4	0	0	1.0000D+00
33	5	2	4	0	1	−4.5858D+00
34	5	2	2	0	3	−8.3400D−01
35	4	1	4	0	0	1.1111D−01
39	4	2	0	0	0	−1.0000D+00
40	4	3	0	0	0	−2.5000D−01
41	4	1	0	0	4	1.9259D+00
46	5	2	0	0	0	0.0000D+00
47	5	3	0	0	0	0.0000D+00
48	5	2	0	0	0	0.0000D+00
52	5	3	0	0	0	5.3266D+00
53	5	3	0	0	5	4.0930D+00
55	6	6	4	0	2	6.3333D+00
56	7	4	0	0	0	−3.4560D+00
60	3	1	0	0	3	3.2568D−02
61	3	2	0	0	0	−1.4365D+02
63	3	2	3	0	0	9.6172D+02
77	5	2	0	0	0	2.4150D−01
78	5	3	0	0	0	−2.9197D+00
79	5	3	0	0	0	7.8777D−02
80	5	3	0	0	5	5.3949D−02
81	5	3	0	0	5	5.3949D−02
111	10	3	0	0	10	−4.7761D+01
112	10	3	10	0	0	−4.7761D+01

In Table 1 we show the data of the selected problems. It shows the name, the dimension of the problem (n), the number of equality constraints (m), the number of variables which are bounded from below (nL), those which are bounded from above (nU), the number of variables which are bounded from below and above ($nL + nU$) and the optimal value of the objective function (f^*). Initial points were the same as in the cited Ref. [33].

5.3. Numerical results

In these numerical experiments we considered that IR-DFO obtains an acceptable solution to a given problem if $\|gf_{tan}^k\| \leq 10^{-5}$, $\|C(x)\| \leq 10^{-5}$ and $r_k \leq 10^{-5}$. The iterative procedure can also finish when the feasibility phase does not satisfy the condition $\|C(y)\| \leq \alpha\|C(x^k)\|$.

We considered that IR-DFO fails to solve a problem in the next three cases:

Fail 1: The number of iterations is greater than 100.
Fail 2: If it is impossible to satisfy $\|C(y)\| \leq \alpha\|C(x^k)\|$ when $\|C(x^k)\| > 10^{-5}$.
Fail 3: The number of evaluations of the objective function (EvalF) is greater than a certain number *fmax*. In these experiments, $fmax = 2000$.

The stopping criterion used in COBYLA is related to trust-region size. In our experiments, the final value for the trust-region bound in COBYLA was set to 10^{-5}.

Firstly, in Table 2 we show the performance of IR-DFOBQA versus IR-DFOTRB with respect to the problems derived from Table 1. It shows the optimal objective value achieved ($f(x_{end})$), the value of the measure of the obtained feasibility ($\|C(x_{end})\|$) and the number of function evaluations with a detail of the number of constraints evaluations (EvalC) and the number of objective function evaluations (EvalF).

We can observe from Table 2 that both versions end with Fail 2 in problems HS34, HS81 and HS111. We can also see no problem ends with failure 3.

The average of evaluations of the objective function required for IR-DFOBQA is 216 while for IR-DFOTRB is 130. Functional values obtained for both implementations are similar. The average evaluations required by IR-DFOBQA in this phase is 467 and 394 by IR-DFOTRB. The obtained values of infeasibility are similar in both versions.

The results of the previous table are also compared to those obtained with the IR method in [28], and to those obtained with COBYLA [32]. The criterion for comparison between those methods is solely based on the number of evaluations of the

Table 2
IR-DFOBQA vs. IR-DFOTRB: feasibility and optimality.

P	IR-DFOBQA			IR-DFOTRB		
HS	EvalC/EvalF	$f(x_{end})$	$\|C(x_{end})\|$	EvalC/EvalF	$f(x_{end})$	$\|C(x_{end})\|$
6	58/78	2.2228D−08	0.0000D+00	61/86	1.1358D−06	0.0000D+00
7	66/175	−1.7321D+00	1.1905D−07	130/166	−1.7320D+00	5.3768D−06
8	129/16	−1.0000D+00	1.7565D−06	96/15	−1.0000D+00	1.6540D−06
9	119/430	−5.0000D−01	1.20486D−06	84/44	−5.0000D−01	1.7256D−06
14	111/21	1.3935D+00	1.6524D−06	144/27	1.3935D+00	3.1641D−06
18	242/17	5.0001D+00	1.43598D−06	308/101	4.9992D+00	4.0675D−06
26	128/67	8.2533D−05	5.3805D−06	148/180	5.1882D−05	5.0994D−07
27	103/88	4.0030D−02	1.6715D−06	91/35	4.0042D−02	1.0094D−08
32	143/22	1.0326D+00	5.0298D−06	398/44	1.0004D+00	1.3941D−06
33	213/23	−4.5858D+00	1.0207D−06	160/56	−4.5853D+00	2.4352D−06
34	542/86	−4.7402D−01	5.9576D−05[a]	969/72	−4.9003D+00	5.2846D−05[a]
35	140/99	1.1375D−01	1.3381D−06	128/38	1.1208D−01	1.7343D−06
39	1440/224	−1.0003D+00	1.8238D−06	1142/122	−1.0000D+00	1.5968D−06
40	194/81	−2.5000D−01	7.3348D−06	247/57	−2.5000D−01	4.3251D−06
41	155/88	1.9299D+00	6.3565D−06	143/56	1.9236D+00	1.7868D−06
46	613/192	1.3863D−04	5.0252D−06	391/105	1.8757D−04	1.3081D−06
47	285/205	1.2274D−04	6.1251D−06	299/205	7.4171D−06	5.5986D−06
48	492/783	1.7550D−06	8.4776D−06	250/91	1.2019D−05	6.1770D−06
52	717/58	5.3267D+00	7.2797D−10	526/310	5.3346D+00	3.1464D−11
53	822/613	4.0913D+00	3.0903D−06	680/464	4.0933D+00	1.5270D−06
55	133/8	6.3333D+00	6.8104D−06	263/9	6.3333D+00	6.1310D−06
56	998/365	−3.4558D+00	1.5403D−06	777/223	−3.4560D+00	7.5157D−06
60	386/110	4.1361D−02	6.5755D−06	151/71	3.2569D−02	2.4060D−06
61	690/138	−1.4318D+02	2.5869D−06	528/138	−1.4360D+02	1.2887D−06
63	449/739	9.6171D+02	2.8318D−06	356/22	9.6171D+02	3.1307D−06
77	301/112	2.4607D−01	1.1790D−06	427/270	2.4154D−01	5.6888D−06
78	150/213	−2.9181D+00	3.9746D−06	396/293	−2.9195D+00	3.6211D−06
79	196/136	7.8840D−02	2.6115D−06	304/216	7.8787D−02	1.0613D−06
80	230/19	5.3567D−02	3.3546D−06	232/20	5.3567D−02	2.3546D−06
81	225/19	5.3565D−02	6.0092D−05[a]	220/21	5.3565D−02	2.3509D−05[a]
111	823/74	−4.7765D+01	5.3176D−05[a]	907/101	−4.7769D+01	5.3994D−05[a]
112	4015/1799	−4.7365D+01	9.5576D−06	1946/600	−4.7376D+01	6.5626D−06

[a] The final solution does not satisfy the optimal value of f or infeasibility measure required.

objective function at the required points. For completeness, in Table 3 we show the numerical results for the IR algorithm and COBYLA.

For these comparisons we use performance profile introduced in [34] and data profile for derivative-free optimization presented in [35]. The performance profile of a solver s is defined as the fraction of problems where the performance ratio is at most α, that is, $\rho_s(\alpha) = \frac{1}{|P|}$ size $\{p : r_{p,s} \leq \alpha\}$, where $r_{p,s} = \frac{t_{p,s}}{\{\min t_{p,s}: s\in S\}}$, $t_{p,s}$ is the number of function evaluations required to satisfy the convergence test, P is the set of problems, $|P|$ denotes the cardinality of P and S is the set of solvers considered.

The data profile of a solver s, which gives the percentage of problems that can be solved with τ function evaluations, is computed by $d_s(\tau) = \frac{1}{|P|}$ size$\{p : t_{p,s} \leq \tau\}$ [35].

The performance profile for the first comparison is shown in Fig. 1. There we compare the results of the two versions of IR-DFO with the results of the IR method of Bueno et al.. We considered the total number of evaluations required (*EvalC* + *EvalF*) in our implementations, while the number of the objective function evaluations is used for IR, according to what was reported by the authors in [28]. Remember also that this method uses derivatives in the restoration phase.

We can notice that the IR-DFOTRB version has done less total evaluations in a 60% of the problems, while IR-DFOBQA in a 50%. Furthermore, the IR method of Bueno et al. has made fewer evaluations of the objective function at approximately 62% problems.

The performance profile in Fig. 1 shows that the IR-DFOTRB proposal has a promising behavior with respect to IR of Bueno et al., since in the latter they use derivatives in the restoration phase.

In the following figures, using the data profiles discussed in [35], we compare IR-DFO with COBYLA analyzing separately the number of objective function evaluations and the number of constraints evaluations as a measure of the performance. In Fig. 2 the performance measure was the number of constraints evaluations. In Fig. 3 the criterion for comparison was the number of objective function evaluations.

In Fig. 2 the data profile shows that COBYLA solves the largest percentage of problems for all sizes less than 1000 evaluations of constraints. We believe that this result is associated to the fact that the two versions of IR-DFO completely recalculated the quadratic model that approximates the function $h(x) = \sum_{i=1}^{m} C_i(x)^2$ each time that the restoration phase is solved.

The data profile of Fig. 3 shows that IR-DFOTRB solves the largest percentage of problems by almost all sizes of the number of objective function evaluations. We can observe that IR-DFOTRB and COBYLA solve 90% of problems with approximately

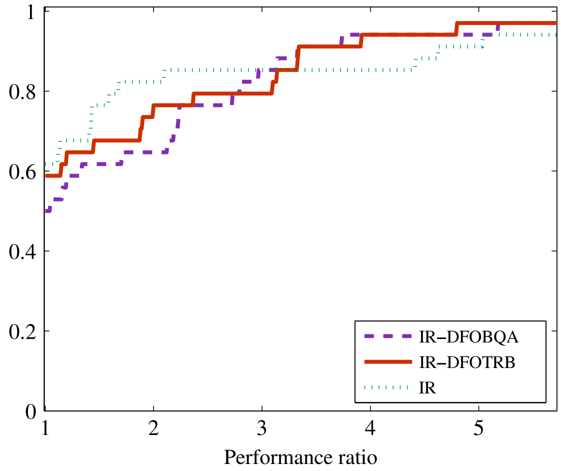

Fig. 1. Performance profile: Total functional evaluations.

Table 3
IR and COBYLA: comparison of the number of evaluations.

P	IR			COBYLA		
HS	EvalF	$f(x_{end})$	$\|C(x_{end})\|$	EvalF[a]	$f(x_{end})$	$\|C(x_{end})\|$
6	366	5.57D−10	4.D−09	46	4.78D−11	5.E−10
7	153	−1.73D+00	1.D−09	51	−1.73D+00	3.D−07
8	4	−1.00D+00	1.D−09	22	−1.00D+00	2.D−07
9	117	−5.00D−01	4.D−15	37	−5.00D−01	5.D−07
14	20	1.39D+00	3.D−09	22	1.39D+00	5.D−08
18	39 217	5.00D+00	9.D−10	105	5.00D+00	6.D−07
26	11 112	1.58D−07	9.D−09	246	1.87D−06	6.D−08
27	4 135	4.00D+00	7.D−09	246	4.00D−02	6.D−09
32	86	1.00D+00	1.D−09	31	1.00D+00	6.E−09
33	54	−4.00D+00[b]	5.D−09	26	−4.58D+00	5.D−05[b]
34	228	−8.34D−01	2.D−09	40	−8.34D−01	3.D−07
35	289	1.11D−01	7.D−10	64	1.11D−01	2.D−10
39	125	−9.89D−01	4.D−09	96	−1.00D+00	4.D−07
40	133	−2.50D−01	2.D−09	77	−2.50D−01	7.D −08
41	430	1.93D+00	1.D−09	69	1.41D+00	0.D+00
46	1 485	1.42D−06	1.D−09	1847	3.93D−06	8.D−08
47	289	1.15D−08	9.D−10	107	1.08D−08	3.D−07
48	861	1.07D−24	2.D−15	91	1.52D−08	6.D−08
52	307	5.33D+00	3.D−09	136	5.33+00	2.D−08
53	308	4.09D+00	3.D−09	99	4.09D+00	6.D−08
55	18	6.67D+00[b]	3.D−09	45	6.66D+00[b]	2.D−07
56	21 267	1.06D−06	5.D−10	232	−3.45D+00	4.D−07
60	596	3.26D−02	8.D−10	53	3.26D−02	8.D−07
61	182	−1.44D+02	3.D−09	90	−8.19D+01[b]	2.D−06
63	171	9.62D+02	4.D−09	59	9.62D+02	6.D−07
77	790	2.41D−01	2.D−09	112	2.41D−01	1.D−06
78	566	−2.92D+00	8.D−09	91	−2.92D+00	5.D−07
79	362	7.88D−02	6.D−09	77	7.87D−02	1.D−06
80	658	5.35D−02	2.D−09	79	5.39D−02	4.D−07
81	770	5.39D−02	6.D−09	113	5.39D−02	2.D−06
111	2 470	−4.28D+01	2.D−09	434	−4.77D+01	6.D−08
112	10 907	−4.78D+01	3.D−09	212	−4.77D+01	2.D−08

[a] In COBYLA the number of constraints evaluations(EvalC) is equal to *EvalF*.
[b] The final solution does not satisfy the optimal value of f or infeasibility measure required.

300 evaluations while IR-DFOBQA approximately 80%. The biggest difference is approximately 10% and it happens when the number of function evaluations is approximately 300.

6. Conclusions

We have presented a new method to solve an optimization problem with general constraints without the use of derivatives. The proposed method is based on Inexact Restoration method [19], which has proven being very successful in nonlinear programming when first order information of the objective function and the constraints is available.

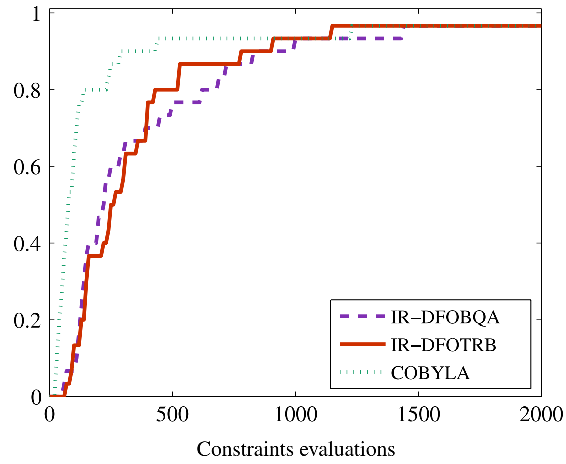

Fig. 2. Data profile: Constraints evaluations.

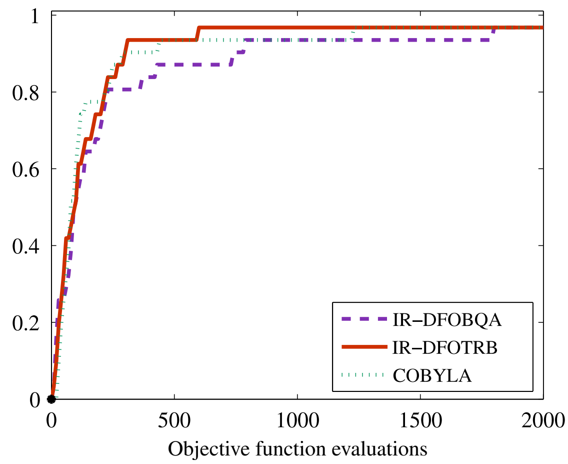

Fig. 3. Data profile: Objective function evaluations.

The complexity, due to the inability to use derivatives, makes getting theoretical results is a difficult task. Therefore, we would like to emphasize that it has been one of the most important parts of this work. Under appropriate assumptions, we have shown the good definition of the algorithm IR-DFO and also convergence to feasible points that satisfy appropriate conditions of optimality.

From the practical point of view, two implementations of the proposed algorithm were tested considering a set of small scale problems. The only difference between the two versions is the algorithm used to solve the Restoration phase. These preliminary implemented versions of IR-DFO have obtained promising numerical results. However, we believe it is necessary to test our algorithm with a more challenging set of problems for more conclusive results. Also, we would like to compare the performance of the tested algorithm with other derivative-free algorithms defined for solving the same problem. It will be necessary to perform a more sophisticated implementation of IR-DFO to improve the procedure to update the sets of interpolation along the iterative process. These will be the subject of our future research.

Since IR-DFO belongs to the class of methods that consider feasibility and optimality at different phases, the user is free to choose different algorithms for both phases. For this purpose different alternatives can be studied to solve each phase. In particular, we would like to define a derivative-free algorithm based on a quadratic model, instead of a linear one, to solve the optimality phase. Moreover, it would be interesting consider filter methods [24–27] instead of merit function to accept or reject the new approximation. Also, it will be a subject of study in our future work.

Acknowledgments

We wish to thank the three anonymous reviewers whose comments and suggestions helped us to improve the contents and readability of our paper.

References

[1] P. Alberto, F. Nogueira, H. Rocha, L.N. Vicente, Pattern search methods for user-provided points: Application to molecular geometry problems, SIAM J. Optim. 14 (2004) 1216–1236.
[2] A. Conn, K. Scheinberg, L.N. Vicente, Introduction to Derivative-Free Optimization, in: MPS SIAM Series on Optimization, SIAM, 2009.
[3] G. Gray, T. Kolda, K. Sale, M. Young, Optimizing an empirical scoring function for transmembrane protein structure determination, INFORMS J. Comput. 16 (2004) 406–418.
[4] M.B. Arouxét, N. Echebest, E.A. Pilotta, Active-set strategy in Powell's method for optimization without derivatives, Comput. Appl. Math. 30 (1) (2011) 171–196.
[5] R. Lewis, V. Torczon, Pattern search algorithms for bound constrained minimization, SIAM J. Optim. 9 (1999) 1082–1099.
[6] M. Marazzi, J. Nocedal, Wedge trust-region methods for derivative-free optimization, Math. Program. 91 (2002) 289–305.
[7] J. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308–313.
[8] M.J.D. Powell, The NEWUOA software for unconstrained optimization without derivatives, in: G. Di Pillo, M. Roma (Eds.), Large-Scale Nonlinear Optimization, Springer, New York, 2006, pp. 255–297.
[9] M.J.D. Powell, The BOBYQA algorithm for bound constrained optimization without derivatives, Cambridge NA Report NA2009/06, University of Cambridge, Cambridge, 2009.
[10] M.A. Abramson, O.A. Brezhneva, J.E. Dennis Jr., R.L. Pingel, Pattern search in the presence of degenerate linear constraints, Optim. Methods Softw. 23 (3) (2008) 297–319.
[11] A.L. Custódio, L.N. Vicente, Using sampling and simplex derivatives in pattern search methods, SIAM J. Optim. 18 (2007) 537–555.
[12] A.L. Custódio, J.E. Dennis, L.N. Vicente, Using simplex gradients of nonsmooth functions in direct search methods, IMA J. Numer. Anal. 28 (2008) 770–784.
[13] T. Kolda, R. Lewis, V. Torczon, Stationarity results for generating set search for linearly constrained optimization, SIAM J. Optim. 17 (2006) 943–968.
[14] R. Lewis, V. Torczon, Pattern search methods for linearly constrained minimization, SIAM J. Optim. 10 (2000) 917–941.
[15] S. Lucidi, M. Sciandrone, P. Tseng, Objective derivative-free methods for constrained optimization, Math. Program. 92 (2002) 37–59.
[16] M.A. Diniz Ehrhardt, J.M. Martínez, L.G. Pedroso, Derivative-free methods for nonlinear programming with general lower-level contraints, Comput. Appl. Math. 30 (2011) 19–52.
[17] R. Lewis, V. Torczon, A globally convergent agmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds, SIAM J. Optim. 12 (2002) 1075–1089.
[18] R. Lewis, V. Torczon, A direct search approach to nonlinear programming problems using an Augmented Lagrangian method with explicit treatment of linear constraints, Technical Report WM-CS-2010-01, College of William and Mary, Department of Computer Sciences, 2010.
[19] J.M. Martínez, E.A. Pilotta, Inexact restoration algorithms for constrained optimization, J. Optim. Theory Appl. 104 (2000) 135–163.
[20] A. Fischer, A. Friedlander, A new line search inexact restoration approach for nonlinear programming, Comput. Optim. Appl. 46 (2010) 336–346.
[21] J.M. Martínez, Inexact-restoration method with lagrangian tangent decrease and new merit function for nonlinear programming, J. Optim. Theory Appl. 111 (2001) 39–58.
[22] J.M. Martínez, E.A. Pilotta, Inexact restoration methods for nonlinear programming: advances and perspectives, in: L. Qi, K. Teo, X. Yang (Eds.), Optimization and Control with Applications, in: Appl. Optim., vol. 96, Springer, New York, 2005, pp. 271–291.
[23] J.M. Martínez, Two-phase model algorithm with global convergence for nonlinear programming, J. Optim. Theory Appl. 96 (1998) 397–436.
[24] C. Gonzaga, E. Karas, M. Vanti, A globally convergent filter method for nonlinear programming, SIAM J. Optim. 14 (3) (2003) 646–669.
[25] C. Gu, A dwindling filter inexact projected Hessian algorithm for large scale nonlinear constrained optimization, Appl. Math. Comput. 219 (2013) 10898–10908.
[26] C. Gu, D. Zhu, A dwindling filter trust region algorithm for nonlinear optimization, Appl. Math. Comput. 240 (2014) 72–81.
[27] E. Karas, E.A. Pilotta, A. Ribeiro, Numerical comparison of merit function with filter criterion in inexact restoration algorithms using hard-spheres problems, Comput. Optim. Appl. 44 (2009) 427–441.
[28] L.F. Bueno, A. Friedlander, J.M. Martínez, F.N. Sobral, Inexact restoration method for derivative-free optimization with smooth constraints, SIAM J. Optim. 23 (2) (2013) 1189–1213.
[29] J.M. Martínez, B.F. Svaiter, A practical optimality condition without constraint qualifications for nonlinear programming, J. Optim. Theory Appl. 118 (2003) 117–133.
[30] D.P. Bertsekas, Nonlinear Programming, second ed., Athena Scientific, Belmont, Massachusetts, 1999.
[31] C. Davis, Theory of positive linear dependence, Amer. J. Math. 76 (1954) 733–746.
[32] M.J.D. Powell, A direct search optimization method t hat models the objective and constraint functions by linear interpolation, in: S. Gomez, J. Hennart (Eds.), Advances in Optimization and Numerical Analysis, Kluwer Academic, Dordrecht, 1994, pp. 51–67.
[33] W. Hock, K. Schittkowski, Test Examples for Nonlinear Programming Codes, in: Lecture Notes in Economics and Mathematical Systems, vol. 187, Springer, 1981.
[34] E. Dolan, J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002) 201–213.
[35] J. Moré, S.M. Wild, Benchmarking derivative-free optimization algorithms, SIAM J. Optim. 20 (1) (2009) 172–191.